

APPLICATION
FOR
UNITED STATES LETTERS PATENT

APPLICANT NAME: Douglas G. Murray

TITLE: METHOD AND SYSTEM VERIFYING PRODUCT
LICENSES USING HARDWARE AND PRODUCT
IDENTIFICATIONS

DOCKET NO.

END9-2000-0164US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence
is being deposited with the United States Postal Service in an
envelope addressed to the Assistant Commissioner for Patents,
Washington, D.C., 20231 as "Express Mail Post Office to Addressee"
Mailing Label No. E1777480466US

on 12/19/00

Denise M. Jurik

Name of person mailing paper

Denise M Jurik
Signature

12/19/00
Date

METHOD AND SYSTEM VERIFYING PRODUCT LICENSES USING HARDWARE AND PRODUCT IDENTIFICATIONS

Background of the Invention

1. Field of the Invention

5 The present invention generally relates to a method and system for verifying software product licenses using hardware and product identifications. More particularly, the present invention provides a method and system for verifying multiple software product licenses in a secure manner using a verification file that is maintained separately from the software products themselves.

2. Background Art

As the use of computer software becomes more prolific, the need to prevent piracy increases. In particular, software producers are constantly faced with growing instances of illegal usages of their products. To help better control the use of programs, most software producers include a license with their products, commonly known as a "shrink-wrap" license. Oftentimes, the license is accompanied by a verification code or key. When installing and/or running the product, an end-user is prompted to input the provided verification code. Upon input of the proper code, the product is consider "verified" and installation/running of the product on the end-user's computer system is

15

20

Summary of the Invention

5 The present invention overcomes the drawbacks of the related art by providing a method and system for verifying product licenses using hardware and product identifications. In addition, the present invention provides a verification file that is obtained/maintained separately from the product itself. Moreover, the verification file of the present invention can contain multiple product and hardware identification so that multiple products can be verified using a single verification file or unit of files. The verification file of the present invention is also encrypted prior to its use to prevent piracy.

According to a first aspect of the present invention, a method for verifying product licenses using hardware and product identifications is provided. The method comprises the steps of: (1) providing a computer hardware component; (2) accessing a verification file, wherein the verification file includes a plurality of product identifications and hardware identifications; and (3) comparing the plurality of hardware identifications with a corresponding hardware identification stored on the hardware component.

20 According to a second aspect of the present invention, a method for verifying product licenses using hardware and product identifications is provided. The method comprises the steps of: (1) providing a processor; (2) installing a verification application on the processor; (3) locating a verification file with the verification application, wherein the verification file includes at least one processor identification and at least one product identification; and (4) comparing

the at least one verification file processor identification with a corresponding processor identification stored on the processor.

According to a third aspect of the present invention, a system for verifying product licenses using hardware and product identifications is provided. The system comprises: (1) an access system for accessing a verification file, wherein the verification file includes a hardware identification and a product identification; (2) a comparison system for comparing the hardware identification of the verification file with a corresponding hardware identification on a computer hardware component; (3) a registry system for registering license information in a registry; and (4) a log system for logging hardware information in a log.

According to a fourth aspect of the present invention, a system for verifying product licenses using hardware and product identifications is provided. The system comprises: (1) a computer hardware component; (2) a verification file having a plurality of hardware identifications and product identifications; (3) an access system for accessing the verification file; and (4) a comparison system for comparing the verification file hardware identifications with a corresponding hardware identification stored on the hardware component, and for comparing the verification file product identifications with a corresponding product identification of a product being installed on the hardware component.

According to a fifth aspect of the present invention, a program product stored on a recordable media for verifying product licenses using hardware and product identifications is provided. When executed the program product

comprises: (1) an access system for accessing a verification file, wherein the verification file include a hardware identification and a product identification; (2) a comparison system for comparing the hardware identification of the verification file with a corresponding hardware identification on a computer hardware component; (3) a registry system for registering license information in a registry; and (4) a log system for logging hardware information in a log.

It is therefore an advantage of the present invention to provide a method and system whereby software products can be verified based upon a specific computer system, as opposed to a specific end-user. It is a further advantage of the present invention to provide a single verifications file (or unit of linked files) for verifying multiple product licenses, which is maintained separately from the programs.

Brief Description of the Drawings

These and other features and advantages of this invention will be more readily understood from the following detailed description of the various aspects of the invention taken in conjunction with the accompanying drawings in which:

Fig. 1 depicts a computer system having a verification application, according to the present invention;

Fig. 2 depicts a logical flow chart of the verification application of Fig. 1;

Fig. 3 depicts a first method flow chart, according to the present invention;

and

Fig. 4 depicts a second method flow chart, according to the present invention.

It is noted that the drawings of the invention are not necessarily to scale. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. The drawings are intended to depict only typical embodiments of the invention, and therefore should not be considered as limiting the scope of the invention. In the drawings, like numbering represents like elements between the drawings.

Detailed Description of the Invention

The present invention generally comprises a method and system for verifying a computer software product based upon a particular computer system, as opposed to a specific end-user. The method and system include a verification file that contains a list of software product identifications and corresponding hardware identifications to indicate the specific computer systems on which installation/running of the programs is permitted. When a product is being installed or run on an end-user's computer system, the verification file is accessed and the identifications in the file are compared with corresponding identifications for the end-user's computer and the product being installed.

Referring now to Fig. 1, a computer system 10 depicting the verification application 30 of the present invention is shown. The computer system 10 generally comprises memory 12, input/output interfaces 14, a central processing

unit (CPU) 16, bus 18 and registry 46. Memory 12 may comprise any known type of data storage and/or transmission media, including magnetic media, optical media, random access memory (RAM), read-only memory (ROM), a data cache, a data object, etc. Moreover, memory 12 may reside at a single physical location, comprising one or more types of data storage, or be distributed across a plurality of physical systems in various forms. CPU 16 may likewise comprise a single processing unit, or be distributed across one or more processing units in one or more locations, e.g., on a client and server.

I/O interfaces 14 may comprise any system for exchanging information from an external source. Bus 18 provides a communication link between each of the components in the server system 10 and likewise may comprise any known type of transmission link, including electrical, optical, wireless, etc. In addition, although not shown, additional components, such as external devices (CRT, LED screen, hand held device, keyboard, mouse, voice recognition system, speech output system, printer, facsimile, pager, personal digital assistant, etc.), cache memory, communication systems, system software, etc., may be incorporated into computer system 10.

Registry 46, as further described below, is a database that stores verification information. Registry 46 may comprise one or more storage devices, such as a magnetic disk drive or an optical disk drive. In another preferred embodiment, registry 46 includes data distributed across, for example, a local area network (LAN), wide area network (WAN) or a storage area network (SAN) (not

utilize conventional token ring connectivity for WAN, LAN, or other private networks, or Ethernet, or other conventional communications standards. Where the client is connected to the system server via the Internet, connectivity could be provided by conventional TCP/IP sockets-based protocol. In this instance, the client would utilize an Internet service provider outside the system to establish connectivity to the system server within the system.

It is understood that the present invention can be realized in hardware, software, or a combination of hardware and software. The computer system 10 according to the present invention can be realized in a centralized fashion in a single computerized workstation, or in a distributed fashion where different elements are spread across several interconnected computer systems (e.g., a network). Any kind of computer system - or other apparatus adapted for carrying out the methods described herein - is suited. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when loaded and executed, controls the computer system 10 such that it carries out the methods described herein. Alternatively, a specific use computer, containing specialized hardware for carrying out one or more of the functional tasks of the invention could be utilized.

The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which - when loaded in a computer system - is able to carry out these methods. Computer program, software program, program, or

software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form.

Referring also now to Fig. 2, a logical flow chart 50 of the method and system of the present invention is depicted. To verify a software product 28, an end-user must first install and run the verification application 52. In a preferred embodiment, the verification application is installed as part of the installation procedure for the product. Specifically, when the end-user attempts to install the product, the verification application 30 will be installed and run on the end-users' computer system. By making the verification application 30 an integral part of product installation, the end-user is forced to verify the product license using the method described herein.

Once installed, the verification application 30 will attempt to access the verification file 22. Verification file 22 contains a list of computer hardware identifications and software program identifications. The hardware identifications correspond to particular end-user computer systems and serve to correlate specific product identifications with a particular computer system. For example, if XYZ, Inc. purchased 10 copies of Microsoft Windows 98®, the verification would include 10 unique product identifications (one for each copy of Windows 98®) and 10 computer hardware identifications that correspond to the specific computer

systems on which each copy of Windows 98® is permitted to be installed. An example of the verification file is as follows:

| <u>PRODUCT IDENTIFICATION</u> | <u>HARDWARE IDENTIFICATION</u> |
|-------------------------------|--------------------------------|
| 123ABC | AB1 |
| 123CDE | AB2 |
| 123EFG | AB3 |
| 123GHI | AB4 |
| 123JKL | AB5 |
| 345ABC | AB6 |
| 345CDE | AB7 |
| 345EFG | AB8 |
| 345GHI | AB9 |
| 345JKL | AB0 |

Thus, each specific product is correlated or linked with a specific end-user's computer system, as opposed to a specific end-user. This prevents the same end-user from installing the same product on multiple computer systems or giving the product to other individuals for installation on their computer systems. The product identification is preferably a unique serial number or verification code assigned to each product by its producer. The hardware identification can be any serial number or the like that is unique to the hardware of a particular computer system. For example, the hardware identification can be a processor identification, a computer serial number, a CD-ROM serial number, etc. It should be understood that although the verification file 22 preferably contains multiple

computer system by inputting the identification for the licensed computer system. However, in this case, the log would be updated to reflect that the particular product has been installed. Thus, the product could never be installed on another computer system (not even the licensed computer system). Although not installed on the licensed computer system, only one installation of the product would be permitted.

Once the log has been updated, the verification application and/or verification file would be removed from the computer system 72 and the verification process is complete 78. Thus, only the installed software product remains on the computer system. It should be appreciated that the verification application could remain on the end-user's computer system if the software producer wishes to verify the product every time the product is executed/run. For example, if an end-user installed Microsoft Windows 98® on his/her computer system, the above-described verification process could be initiated at every start-up. If it is determined that the product is not legally installed on the computer system (e.g., the license has expired), the entire product could be removed by removal system 42.

Referring now to Fig. 3 a flow chart of a first method 100 according to the present invention is shown. As depicted, the first step 102 of the method 100 is to provide a computer hardware component. The second step 104 is to access a verification file, wherein the verification file includes a plurality of product identifications and hardware identifications. The third step 106 of method 100 is

to compare the plurality of hardware identifications with a corresponding hardware identification stored on the hardware component.

Referring to Fig. 4, a flow chart of a second method 200 according to the present invention is shown. The first step 202 of method 200 is to provide a processor. The second step 204 is to install a verification application on the processor. The third step 206 of method 200 is to locate a verification file with the verification application, wherein the verification file includes at least one processor identification and at least one product identification. The fourth step 208 is to compare the at least one verification file processor identification with a corresponding processor identification stored on the processor.

The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously, many modifications and variations are possible. Such modifications and variations that may be apparent to a person skilled in the art are intended to be included within the scope of this invention as defined by the accompanying claims.